

IN THE CLAIMS:

- 1 1. (Previously Presented) A method for striping packets across pipelines of a
2 processing engine within a network switch, the processing engine having a plurality of
3 processors arrayed as pipeline rows and columns embedded between input and output
4 buffers, the method comprising the steps of:
5 including a context memory in each pipeline row;
6 organizing the context memory as a plurality of window buffers of a defined size;
7 apportioning each packet into contexts corresponding to the defined size associ-
8 ated with each window buffer; and
9 correlating each context with a relative position within the packet to thereby fa-
10 cilitate reassembly of the packet at the output buffer, while obviating out-of-order issues
11 involving the contexts of the packet.

- 1 2. (Original) The method of Claim 1 further comprising the step of organizing
2 the processors and context memory of each pipeline row as a cluster.

- 1 3. (Original) The method of Claim 2 wherein the step of apportioning comprises
2 the steps of:
3 segmenting the packet into fixed sized contexts at the input buffer;
4 sequentially passing the contexts to the clusters; and

5 storing the fixed sized contexts in appropriate window buffers of the context
6 memories.

1 4. (Original) The method of Claim 3 wherein the step of correlating comprises
2 the step of providing a program counter entry point function to indicate the relative posi-
3 tion of each context within the packet.

1 5. (Original) The method of Claim 3 wherein the relative position comprises one
2 of a beginning, middle and end context of the packet.

1 6. (Original) The method of Claim 3 further comprising the steps of:
2 processing the context at a source processor of the cluster;
3 communicating an intermediate result relating to processing of the context to a
4 destination processor of a neighboring cluster.

1 7. (Original) The method of Claim 6 wherein the step of communicating com-
2 prises the step of providing an intercolumn communication mechanism configured to
3 forward the intermediate result from the source processor to an address of the destination
4 processor.

1 8. (Original) The method of Claim 3 further comprising the step of changing the
2 size of a fixed sized context at the context memory of a cluster.

1 9. (Currently Amended) [[The method of Claim 8 wherein the step of changing
2 comprises the steps of:]]

3

4 A method for striping packets across pipelines of a processing engine within a network
5 switch, the processing engine having a plurality of processors arrayed as pipeline rows
6 and columns embedded between input and output buffers, the method comprising the
7 steps of:

8 including a context memory in each pipeline row;

9 organizing the context memory as a plurality of window buffers of a defined size;
10 apportioning each packet into contexts corresponding to the defined size associ-
11 ated with each window buffer by,

12 segmenting the packet into fixed sized contexts at the input buffer;

13 sequentially passing the contexts to the clusters; and

14 storing the fixed sized contexts in appropriate window buffers of the con-
15 text memories;

16 correlating each context with a relative position within the packet to thereby fa-
17 cilitate reassembly of the packet at the output buffer, while obviating out-of-order issues
18 involving the contexts of the packet;

19 organizing the processors and context memory of each pipeline row as a cluster;

20 changing the size of a fixed sized context at the context memory of a cluster;

21 deleting a portion of the fixed sized context stored in the window buffer; and

22 substituting the deleted portion of the context with information stored at another
23 location of the context memory.

1 10. (Original) The method of Claim 9 wherein the substituted information is one
2 of larger than and smaller than the deleted portion of the fixed sized context.

1 11. (Original) A system for striping packets across pipelines of a processing en-
2 gine within a network switch, the processing engine having a plurality of processors ar-
3 rayed as pipeline rows and columns embedded between input and output buffers, the sys-
4 tem comprising:

5 a context memory within each pipeline row, the context memory organized as a
6 plurality of window buffers of a defined size;

7 a segmentation unit adapted to apportion each packet into contexts for processing
8 by the processors, each context corresponding to the defined size associated with each
9 window buffer; and

10 a mapping mechanism configured to correlate each context with a relative posi-
11 tion within the packet to thereby facilitate reassembly of the packet at the output buffer,
12 while obviating out-of-order issues involving the contexts of the packet.

1 12. (Original) The system of Claim 11 wherein the processors and context mem-
2 ory of each pipeline row are organized as a cluster.

1 13. (Original) The system of Claim 12 wherein the mapping mechanism com-
2 prises a program counter entry point function that indicates the relative position of each
3 context within the packet.

1 14. (Original) The system of Claim 13 wherein the relative position comprises
2 one of a first, last and intermediate portion of the packet.

1 15. (Original) The system of Claim 13 further comprising an intercolumn com-
2 munication mechanism configured to forward an intermediate result relating to process-
3 ing of a context by a source processor to a destination processor.

1 16. (Original) A computer readable medium containing executable program in-
2 structions for striping packets across pipelines of a processing engine within a network
3 switch, the processing engine having a plurality of processors arrayed as pipeline rows
4 and columns embedded between input and output buffers, each pipeline row including a
5 context memory, the processors and context memory of each pipeline row organized as a
6 cluster, the executable program instructions comprising program instructions for:
7 organizing the context memory as a plurality of window buffers of a defined size;
8 apportioning each packet into contexts corresponding to the defined size associ-
9 ated with each window buffer; and

10 correlating each context with a relative position within the packet to thereby fa-
11 cilitate reassembly of the packet at the output buffer, while obviating out-of-order issues
12 involving the contexts of the packet.

1 17. (Original) The computer readable medium of Claim 16 further comprising
2 program instructions for:
3 segmenting the packet into fixed sized contexts at the input buffer;
4 sequentially passing the contexts to the clusters; and
5 storing the fixed sized contexts in appropriate window buffers of the context
6 memories.

1 18. (Original) The computer readable medium of Claim 17 wherein the program
2 instruction for correlating comprises the program instruction for providing a program
3 counter entry point function to indicate the relative position of each context within the
4 packet.

1 19. (Original) The computer readable medium of Claim 17 further comprising
2 program instructions for changing the size of a fixed sized context at the context memory
3 of a cluster.

20. (Currently Amended) [[The computer readable medium of Claim 19 wherein
3 the program instruction for changing comprises program instructions for:]]

4 A computer readable medium containing executable program instructions for striping
5 packets across pipelines of a processing engine within a network switch, the processing
6 engine having a plurality of processors arrayed as pipeline rows and columns embedded
7 between input and output buffers, each pipeline row including a context memory, the
8 processors and context memory of each pipeline row organized as a cluster, the executa-
9 ble program instructions comprising program instructions for:

10 organizing the context memory as a plurality of window buffers of a defined size;
11 apportioning each packet into contexts corresponding to the defined size associ-
12 ated with each window buffer;

13 correlating each context with a relative position within the packet to thereby fa-
14 cilitate reassembly of the packet at the output buffer, while obviating out-of-order issues
15 involving the contexts of the packet;

16 segmenting the packet into fixed sized contexts at the input buffer;
17 sequentially passing the contexts to the clusters;
18 storing the fixed sized contexts in appropriate window buffers of the context
19 memories;
20 changing the size of a fixed sized context at the context memory of a cluster;
21
22 deleting a portion of the fixed sized context stored in the window buffer; and

23 substituting the deleted portion of the context with information stored at another
24 location of the context memory.

1 21. (Previously Presented) Electromagnetic signals propagating on a computer
2 network carrying instructions for striping packets across pipelines of a processing engine
3 within a network switch, the processing engine having a plurality of processors arrayed as
4 pipeline rows and columns embedded between input and output buffers, each pipeline
5 row including a context memory, the processors and context memory of each pipeline
6 row organized as a cluster, the electromagnetic signals comprising program instructions
7 for:
8 organizing the context memory as a plurality of window buffers of a defined size;
9 apportioning each packet into contexts corresponding to the defined size associ-
10 ated with each window buffer; and
11 correlating each context with a relative position within the packet to thereby fa-
12 cilitate reassembly of the packet at the output buffer, while obviating out-of-order issues
13 involving the contexts of the packet.

Please add new Claims 22 *et al.*, as follows:

1 22. (New) A method for operating a network switch, comprising:
2 arraying a plurality of processors as a plurality of rows, a row forming a pipeline
3 row, the pipeline rows arrayed between an input buffer and an output buffer;
4 including a context memory in each pipeline row;
5 organizing the context memory as a plurality of window buffers of a defined size;
6 apportioning, by the input buffer, each packet into packet contexts, a packet con-
7 text corresponding to the defined size associated with each window buffer; and
8 correlating each packet context with a relative position within the packet to
9 thereby facilitate reassembly of the packet at the output buffer, to facilitates striping
10 packets across a plurality of the pipelines.

1 23. (New) The method of claim 22, further comprising:
2 organizing the processing engine so that a pipeline row forms a cluster of proces-
3 sors, and including the context memory as part of the cluster.

1 24. (New) The method of claim 22, further comprising:
2 segmenting the packet into fixed sized contexts at the input buffer;
3 sequentially passing the contexts to the clusters; and
4 storing the fixed sized contexts in appropriate window buffers of the context
5 memories.

1 25. (New) The method of claim 22, further comprising:
2 providing a program counter entry point function to indicate the relative position
3 of each context within the packet.

1 26. (New) The method of claim 22, wherein the relative position further com-
2 prises:
3 one of a beginning, middle and end context of the packet.

1 27. (New) The method of claim 22, further comprising:
2 processing the context at a source processor of the cluster;
3 communicating an intermediate result relating to processing of the context to a
4 destination processor of a neighboring cluster.

1 28. (New) The method of claim 27, further comprising:
2 providing an intercolumn communication mechanism configured to forward the
3 intermediate result from the source processor to an address of the destination processor.

1 29. (New) The method of claim 22, further comprising:
2 changing the size of a fixed sized context at the context memory of a cluster.

1 30. (New) The method of claim 22, further comprising:

2 deleting a portion of the fixed sized context stored in the window buffer; and
3 substituting the deleted portion of the context with information stored at another
4 location of the context memory.

1 31. (New) A processing engine within a network switch, comprising:
2 means for arraying a plurality of processors as a row of a plurality of pipelines,
3 the rows arrayed between an input buffer and an output buffer;
4 means for including a context memory in each pipeline row;
5 means for organizing the context memory as a plurality of window buffers of a
6 defined size;
7 means for apportioning, by the input buffer, each packet into packet contexts, a
8 packet context corresponding to the defined size associated with each window buffer; and
9 means for correlating each packet context with a relative position within the
10 packet, to facilitate reassembly of the packet at the output buffer, thereby facilitating
11 striping packets across the plurality of pipelines.

1 32. (New) The processing engine of claim 31, further comprising:
2 means for organizing the processing engine so that a pipeline row forms a cluster
3 of processors, and including the context memory as part of the cluster.

1 33. (New) The processing engine of claim 31, further comprising:
2 means for segmenting the packet into fixed sized contexts at the input buffer;

3 means for sequentially passing the contexts to the clusters; and
4 means for storing the fixed sized contexts in appropriate window buffers of the
5 context memories.

1 34. (New) The processing engine of claim 31, further comprising:
2 means for providing a program counter entry point function to indicate the rela-
3 tive position of each context within the packet.

1 35. (New) The processing engine of claim 31, wherein the relative position fur-
2 ther comprises:
3 means for one of a beginning, middle and end context of the packet.

1 36. (New) The processing engine of claim 31, further comprising:
2 means for processing the context at a source processor of the cluster;
3 means for communicating an intermediate result relating to processing of the con-
4 text to a destination processor of a neighboring cluster.

1 37. (New) The processing engine of claim 36, further comprising:
2 means for providing an intercolumn communication mechanism configured to
3 forward the intermediate result from the source processor to an address of the destination
4 processor.

1 38. (New) The processing engine of claim 31, further comprising:
2 means for changing the size of a fixed sized context at the context memory of a
3 cluster.

1 39. (New) The processing engine of claim 31, further comprising:
2 means for deleting a portion of the fixed sized context stored in the window
3 buffer; and
4 means for substituting the deleted portion of the context with information stored
5 at another location of the context memory.

1 40. (New) A processing engine within a network switch, comprising:
2 a plurality of processors arrayed as a plurality rows, a row forming a pipeline, the
3 plurality of processors forming a plurality of rows of pipelines, the rows arrayed between
4 an input buffer and an output buffer;
5 a context memory included in each pipeline row;
6 the context memory organized as a plurality of window buffers of a defined size;
7 the input buffer apportioning each packet into packet contexts, a packet context
8 corresponding to the defined size associated with each window buffer; and
9 a processor of the plurality of processors to correlate each packet context with a
10 relative position within the packet, to facilitate reassembly of the packet at the output
11 buffer, thereby facilitating striping packets across the plurality of pipelines.

1 41. (New) The processing engine of claim 40, further comprising:
2 the processing engine organized so that a pipeline row forms a cluster of proces-
3 sors, and including the context memory as part of the cluster.

1 42. (New) The processing engine of claim 40, further comprising:
2 the input buffer segmenting the packet into fixed sized contexts;
3 the input buffer sequentially passing the contexts to the clusters; and
4 window buffers to store the fixed sized contexts in appropriate context memories.

1 43. (New) The processing engine of claim 40, further comprising:
2 a program counter entry point function to indicate the relative position of each
3 context within the packet.

1 44. (New) The processing engine of claim 40, wherein the relative position fur-
2 ther comprises:
3 a beginning, middle and end context of the packet.

1 45. (New) The processing engine of claim 40, further comprising:
2 a source processor of the cluster to process the context; and
3 a destination processor of a neighboring cluster to receive an intermediate result
4 relating to processing of the context.

1 46. (New) The processing engine of claim 45, further comprising:
2 an intercolumn communication mechanism configured to forward the intermedi-
3 ate result from the source processor to an address of the destination processor.

1 47. (New) The processing engine of claim 40, further comprising:
2 a processor of the plurality of processors to change the size of a fixed sized con-
3 text at the context memory of a cluster.

1 48. (New) The processing engine of claim 40, further comprising:
2 a processor of the plurality of processors to change the size of a fixed sized con-
3 text at the context memory of a cluster.